Spatial-temporal Graph Transformer Network for Spatial-temporal Forecasting*

Duy Tang Hoang[0000-0002-3930-7283], Minh Son Dao[0000-0003-3044-8175], and Koji Zettsu[0000-0003-4062-2376]

National Institute of Information and Communications Technology (NICT)
4-2-1, Nukui-Kitamachi, Koganei, Tokyo 184-8795, Japan
{hoang.duy.tang, dao, zettsu}@nict.go.jp

Abstract. Spatial-temporal data analysis and forecasting pose significant challenges due to the complex interplay of spatial and temporal dependencies. Traditional methods often struggle to effectively capture these dynamics. In this study, we propose a novel approach to address these challenges—a dual attention mechanism graph transformer. This model leverages both spatial and temporal information to better capture the intricate patterns inherent in spatial-temporal data. Experimental evaluations are conducted on two distinct forecasting tasks: air pollution forecasting and traffic flow forecasting. The results demonstrate the superior performance of the proposed graph transformer compared to other graph neural networks. The proposed model represents an advancement in spatial-temporal data analysis and forecasting. By incorporating a dual attention mechanism, our model effectively captures the complex relationships within spatial-temporal data, leading to improved forecasting performance. This approach holds promise for various applications requiring accurate forecasting in spatial-temporal domains.

Keywords: Graph Transformer, Graph Data, Spatial-temporal Data, Air Pollution Forecasting, Traffic Flow Forecasting.

1 Introduction

Spatial-temporal forecasting refers to the prediction of future values or trends of a variable that varies both spatially and temporally. In other words, it involves forecasting how a phenomenon changes over both space (location) and time. This type of forecasting is particularly relevant in fields where data is collected across multiple spatial locations at different time intervals, such as meteorology, environmental science, transportation, epidemiology, and economics. For example, in air pollution forecasting, spatial-temporal forecasting involves predicting pollutant concentrations at various locations over time. Similarly, in traffic flow forecasting, it involves predicting traffic congestion levels at different locations

* The conceptualization and design of the Spatial-temporal Graph Transformer Network were carried out by Minh Son Dao.

and time periods. Spatial-temporal forecasting techniques typically take into account both the spatial dependencies (relationships between different locations) and temporal dependencies (relationships between different time points) in the data.

Spatial-temporal forecasting presents a lot of challenges originate from the complexities of dynamic systems evolving over both space and time [12]. One significant difficulty arises from the interplay between spatial and temporal dependencies within the data. Capturing these dependencies accurately requires models capable of discerning complex patterns and relationships across multiple dimensions simultaneously. Additionally, spatial-temporal data often exhibit non-linear and non-stationary behaviors, further complicating the forecasting task. The heterogeneity of data across different spatial locations and time intervals introduces additional sources of variability, posing challenges for model generalization. Moreover, spatial-temporal forecasting may encounter issues related to data sparsity, irregular sampling, and missing observations, necessitating robust techniques for data pre-processing and imputation. Furthermore, the dynamic nature of real-world systems introduces uncertainties and external factors that can influence future outcomes, making forecasting more challenging. Addressing these difficulties requires the development of advanced modeling approaches, innovative techniques for feature extraction and representation, and effective strategies for incorporating domain knowledge and expert insights into the forecasting process.

Traditional methods for spatial-temporal forecasting typically rely on statistical approaches and time-series analysis techniques. These methods often involve dis-aggregating the spatial and temporal dimensions of the data and treating them separately. For spatial forecasting, methods such as spatial autocorrelation models [20] and spatial interpolation techniques [1, 2] are commonly used to capture spatial dependencies and interpolate values between observation points. Temporal forecasting, on the other hand, typically involves time-series analysis methods such as auto-regressive integrated moving average (ARIMA) models [9, 7], and Auto-regression (VAR) and its derivatives, including VARMA, VARMAX, and SVAR [3, 5]. The challenge arises as VAR-based techniques necessitate stationary in all involved time series, rendering them unsuitable for the context of real-world tasks like air pollution forecasting or traffic flow forecasting where data exhibits non-stationary behavior.

Recent studies have demonstrated the significant potential of deep learning approaches to enhance the accuracy and efficiency of spatial-temporal forecasting [14, 19, 15, 6, 11]. Recent research has shown a significant interest in graph-based neural networks, as their structure inherently possesses the capability to model graph data, which closely resembles the topology of multiple air pollution stations. With the design aimed at capturing both temporal and spatial features of data using graph neural networks, researchers have developed various types of spatial-temporal graph neural networks. B. Yu et al. [6] introduced STGCN, which comprises multiple spatial-temporal convolutional blocks. Each block is structured with two gated sequential convolution layers, sandwiching a spatial graph convolutional layer. M. Xu et al. [11] proposed a graph neural network that integrates a transformer architecture, featuring two distinct components: a spatial transformer and a temporal transformer. H. Liu et al. [18] devised a sophisticated model comprising two distinct modules: a spatial-temporal relationship interaction module and a spatial-temporal feature extraction module. K. Gong et al. [16] developed a model tailored for traffic data, incorporating adaptive graph convolution layers and the Sample Convolution and Interaction Network (TrafficSCINet) [13]. Through the literature review of these models, it is evident that a common approach involves integrating various architectures and mechanisms, such as Convolution, Attention, and Transformer, into the design, with the aim of enabling the model to capture the intricate spatial-temporal dynamics of the data.

In this study, we present a novel graph learning model designed to address the challenges of spatial-temporal forecasting. Central to our proposed model is the Spatial-Temporal Graph Transformer (STGT) layer, featuring a dual attention mechanism. Multiple STGT layers are then stacked together to construct an encoder-decoder architecture-like model, named Spatial-temporal Graph Transformer Network (STGTN), facilitating forecasting future time steps in an auto-regressive manner. To evaluate the effectiveness of the proposed model, experimental study is conducted with two real spatial-temporal datasets including an air pollution dataset and a traffic flow dataset. Overall, the main contribution of this work are as follows.

- A Spatial-Temporal Graph Transformer (STGT) layer is proposed, featuring a unique dual attention mechanism.
- A Spatial-Temporal Graph Transformer Network (STGTN) with encoder-decode structure consisting of multiple STGT units is designed to address the forecasting task in an auto-regressive manner.
- An experimental study is conducted using real-world datasets, including air pollution forecasting and traffic flow forecasting tasks, with the results demonstrating the superiority of the proposed model.

2 Problem Formulation

Spatial-temporal forecasting aims to predict future values or trends of a target variable that varies both spatially and temporally. In this section, we formally define the problem and outline the key components involved in spatial-temporal forecasting.

The target variables represents the phenomenon of interest that is observed at multiple spatial locations and time intervals. Examples include air pollutant concentrations, traffic flow rates, weather parameters, and disease incidence rates.

A spatial-temporal dataset consists of spatial domain and temporal domain. The spatial domain consists of a set of N distinguish locations where the target variables are observed. Each location is characterized by its geographical coordinates (latitude and longitude) and may have associated attributes such as elevation, land use, and population density. The temporal domain T comprises a sequence of discrete time intervals or timestamps at which observations of the target variable are recorded. The time intervals may range from minutes or hours to days, weeks, months, or years, depending on the specific application.

Accordingly, the spatial-temporal dataset acquired over a time period from multiple sites (monitoring stations), can be represented in the form a multi-dimension data array $\mathbf{A} \in {\mathbb{R}_+}^{M \times N \times P}$, where N is the number of monitoring stations, M and P are inherited from A. The forecasting involves in using the historical observations of air pollutants to estimate the concentration values of the corresponding pollutants in the future. This process can be described as in Equation 1 as follows:

$$Y = \mathbb{F}(X) \tag{1}$$

where $Y \in {\mathbb{R}_+}^{F \times N \times P}$ is the predicted values; $X \in {\mathbb{R}_+}^{H \times N \times P}$ is the historical observations; H and F are the number of historical time steps and future time steps, respectively; $\mathbb{F}$ denotes the forecasting function.

3 Methodology

3.1 Graph Representation of Spatial-temporal Data

Consider a spatial-temporal dataset with the spatial domain consist of N locations. The spatial relationship of the dataset can be represented by a graph $G = (V, E, A)$, where V, E, and A are the set of vertices, the set of edges and the adjacency matrix, respectively. In the graph, each vertex is depicted by a location. The edges between vertices are determined by specific constrain, such as geographical distance. Consider in the whole time domain, we have a set of different states $\{G_1, G_2, .., G_t\}$, where G_i indicates the state of graph G at time i. The forecasting problem in Equation 1 can be considered as using H historical graph states to predict F future graph states as follows:

$$G_Y = \mathbb{F}(G_X) \tag{2}$$

3.2 Spatial-Temporal Graph Transformer Layer

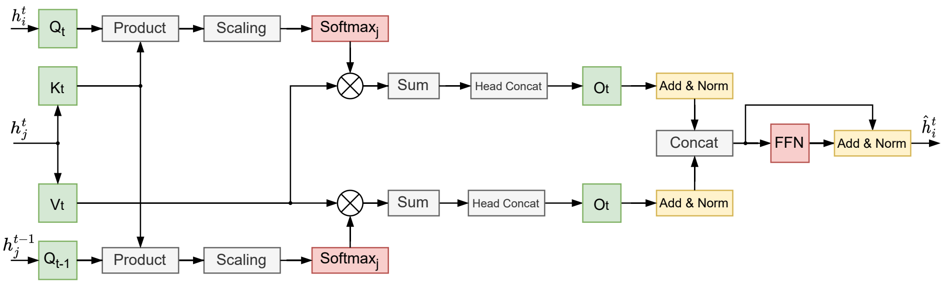

Fig. 1. Spatial-temporal graph transformer layer

We expand the graph transformer layer introduced in [10] by introducing a dual attention mechanism as illustrated in Figure 2. In addition to calculating the attention weights between a vertex and its neighbors at the current time step, our expanded layer enables a vertex to reuse its embedding at the previous time step as a query vector. Consequently, there are two query vectors at a vertex at each time step (i.e., the vertex's embedding at both the current and previous frames). Figure 1 illustrates our proposed layer in detail. According to the figure, the module consists of two parallel multi-head dot-product attention modules. The results are concatenated at the top of two modules and passed onto a feed-forward network to produce the final output. It is worth noting that both queries access shared keys and values. Key and value vectors are the neighbors' information, which represents the traffic conditions at the current time step. This setup allows the layer to learn both spatial and temporal correlation directly.

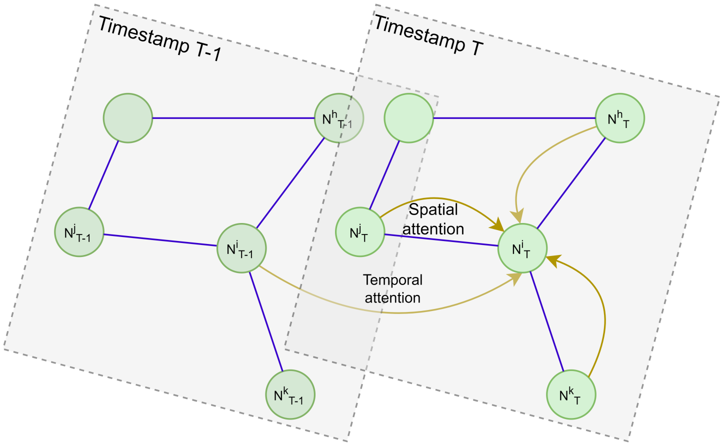

Fig. 2. Dual attention

Let $\{h_i^t\}, h_i^t \in \mathbb{R}^d$ be the set feature vectors of N vertices in the graph at a time step t. We define the dot-product attention weights at each vertex as follows:

$$w_{i,j}^t = \text{softmax}_j\Big(\frac{Q_t h_i^t \cdot K_t h_j^t}{\sqrt{d_k}}\Big) \tag{3}$$

$$w_{i,j}^{t-1} = \text{softmax}_j\Big(\frac{Q_{t-1} h_i^{t-1} \cdot K_t h_j^t}{\sqrt{d_k}}\Big) \tag{4}$$

$Q_t, Q_{t-1} \in \mathbb{R}^{d_k \times d}$ denote the query projection matrices for two separate input vectors (i.e., current and previous embeddings of a vertex). $K_t, V_t \in \mathbb{R}^{d_k \times d}$ are key and value projection matrices, respectively. Softmax operations are performed among the neighbors $\{h_j\}$ of h_i. Then, two hidden embeddings are correspondingly calculated as:

$$\bar{h}_i^t = O_t \mathbin{\Vert}_{k=1}^{H} \Big(\sum_{j \in \mathbf{N}_i} w_{i,j}^t V_t h_j^t \Big) \tag{5}$$

$$\bar{h}_i^{t-1} = O_{t-1} \mathbin{\Vert}_{k=1}^{H} \Big(\sum_{j \in \mathbf{N}_i} w_{i,j}^{t-1} V_t h_j^t \Big) \tag{6}$$

$O_t,\ O_{t-1} \in \mathbb{R}^{d \times d}$ are two projection matrices. $\Vert_{k=1}^{H}$ is the concatenation of H heads in the transformer layer. Two results are added with their residuals and normalized as

$$\tilde{h}_i^t = \mathrm{Norm}(\bar{h}_i^t + h_i^t) \tag{7}$$

$$\tilde{h}_i^{t-1} = \mathrm{Norm}(\bar{h}_i^{t-1} + h_i^{t-1}) \tag{8}$$

Two attention outputs are concatenated and preceded to a feed-forward network. The final output is produced by adding residuals into the network's output and normalizing the output tensor. The formulas are written as follows:

$$h_i'^t = \tilde{h}_i^t \mathbin{\Vert} \tilde{h}_i^{t-1} \tag{9}$$

$$h_i''^t = F\Big(h_i'^t\Big) \tag{10}$$

$$\hat{h}_i^t = \mathrm{Norm}\Big(h_i''^t + h_i'^t\Big) \tag{11}$$

F denotes a feed-forward network with a number of layers and activation. The final output $\hat{h}_i^t$ is the updated representation for a vertex at the current time step. Notably, $\hat{h}_i^t$ can be used as the temporal embedding input for the next time step.

3.3 Spatial-temporal Graph Transformer Network

To addressing the forecasting task in a auto-regression manner, multiple STGT layers are stacked together to construct a encoder - decoder structure as in Figure 3. In the encoder, each input sample is preceded to a linear embedding layer to produce high-dimension embedding vectors. $\lambda \in \mathbb{R}^{N \times k}$ indicates the spatial positional encoding introduced in [10]. It is the Laplacian eigenvector of the graph and can be pre-computed. N is the number of vertices and k is the number of smallest non-zero eigenvalues. λ is passed to a linear layer to produce vectors that have the same dimension with the input embeddings. We can write the addition as follows:

$$\tilde{h}_i^t = W_1 X_i^t + W_2 \lambda_i \tag{12}$$

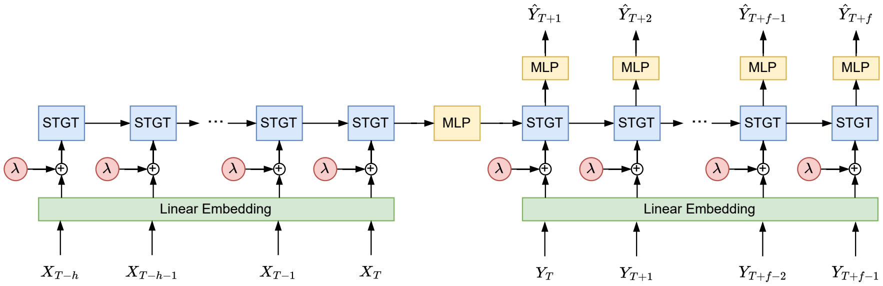

Fig. 3. Spatial-temporal Graph Transformer Network

The equation only shows the addition at a certain vertex for simplicity. In the equation, X_i^t denotes the input sample of the vertex i at time step t and λ_i is the corresponding positional encoding vector at this vertex. $W_1 \in \mathbb{R}^{d_k \times 1}$ and $W_2 \in \mathbb{R}^{d_k \times k}$ are two transformation matrices mapping the two vectors into spaces that have the same dimension. Afterwards, the addition results (i.e., $\{\tilde{h}_i^t\}$) are passed to a STGT layer. The number of STGT layers varies correspondingly with the length of the sequence. In particular, the encoder comprises H layers of STGT, and each of them corresponds to a certain time step in the input sequence. As mentioned above, we incorporate the output of the previous STGT layer (i.e., $\{\hat{h}_i^{t-1}\}$ from previous time step) into that of the current time step. Finally, the last output of the encoder is preceded to a multi-layer perceptron (MLP), resulting in a refined representations.

Similar to the encoder's architecture, the decoder also comprises a sequence of STGT layers and a linear embedding layer. On the top of the STGT layers, there is a MLP to produce the final prediction at each time step. The module made the temporal predictions step by step. The final output at each time step is computed as:

$$\hat{Y}_{t+1} = \text{MLP}(L_t(\tilde{h}_t)) \tag{13}$$

where L_t denotes the t^{th} STGT layer in the decoder, $\tilde{h}_t$ is computed by the Equation 12. $\hat{Y}_{t+1} \in \mathbb{R}^{N \times 1}$ is the prediction for all vertices of the next time step. Then, the result can be used as the input for the next step. The prediction process stops after T steps. Thus, there are F future time steps to be predicted in our model. In addition, our proposed architecture can be used for either single-step or multiple-step prediction task. F can be set as 1 in the former setup, while $F > 1$ for the latter scenario.

4 Experimental Study

4.1 Air Pollution Dataset

The air pollutant dataset in this work was measured from 17 monitoring stations in Chiba, Japan, over a three-year period, spanning from April 1$^{\text{st}}$ to September

30th in the years 2017, 2018, and 2019. The dataset encompasses hourly concentration measurements of two air type of air pollutants, nitrogen oxides (NO_x) and oxidants (O_x), with units expressed in parts per billion (ppb).

Figures 4 and 5 depict the daily and hourly averages of the dataset, respectively. A distinct pattern emerges from both visuals, illustrating that the concentration of NO_x is notably lower compared to that of O_x. Furthermore, it is evident that the daily trend exhibits significant fluctuations, whereas the hourly trend appears considerably smoother. Although Figure 4 doesn't exhibit a discernible temporal pattern, Figure 5 illustrates a distinct hourly trend that repeats consistently across the three-year period spanning 2017, 2018, and 2019. For pollutant NO_x, the period from 13:00 to 15:00 consistently registers the highest concentrations. Conversely, in the case of O_x, the peak concentrations occur during a different time frame, around 9:00.

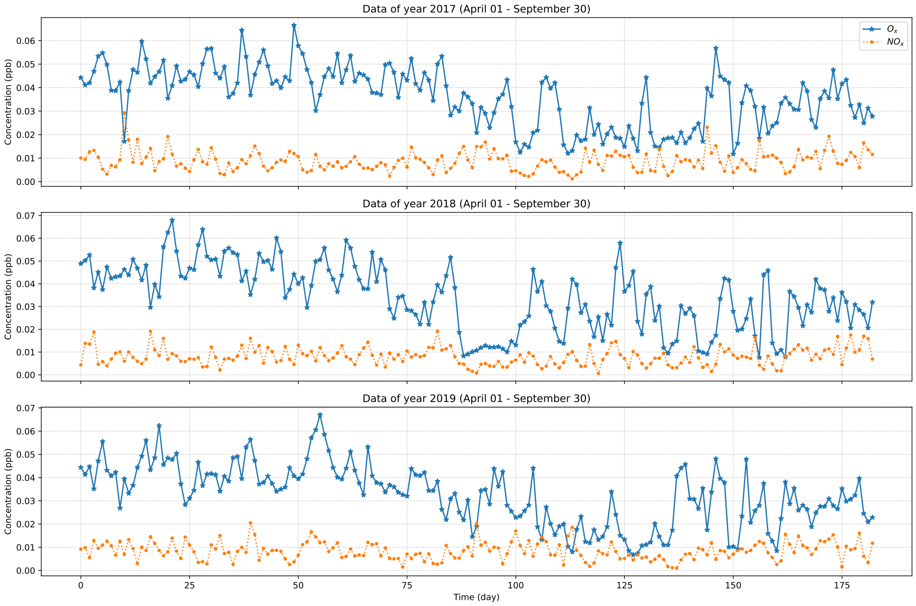

Fig. 4. Daily concentration

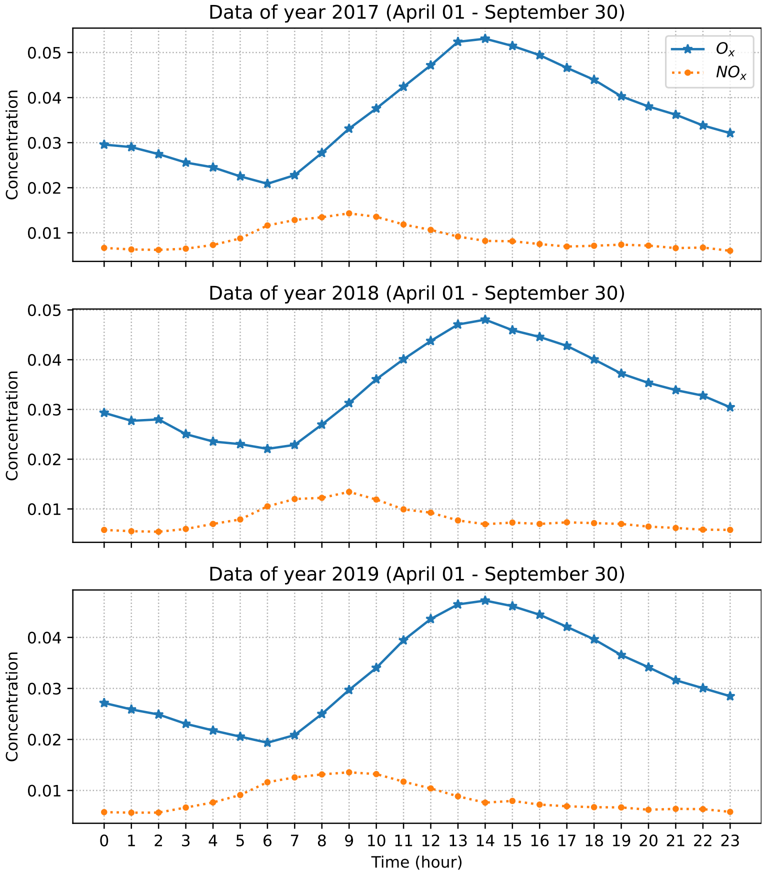

Fig. 5. Hourly concentration

Anomaly Removal: A anomaly removal procedure was implemented to enhance the quality and reliability of the dataset. Specifically, for the nitrogen oxides (NO_x) time series data, the $mean$ and standard deviation (std) were calculated to characterize the typical behavior of the pollutant. Any observations surpassing the threshold of $mean + 12 * std$ the standard deviation were identified as anomalies and subsequently removed from the dataset. This rigorous criterion ensured the exclusion of extreme values that could potentially distort the analysis. Similarly, for the (O_x) time series data, a slightly less stringent threshold was applied. The mean and standard deviation were computed, and observations exceeding the threshold of $mean + 5 * std$ were treated as anomalies and excluded from the dataset. This tailored approach to anomaly removal aimed to strike a balance between preserving relevant information and mitigating the influence of outliers in the air pollution time series data. The results of anomaly removal step are shown in Figure 6.

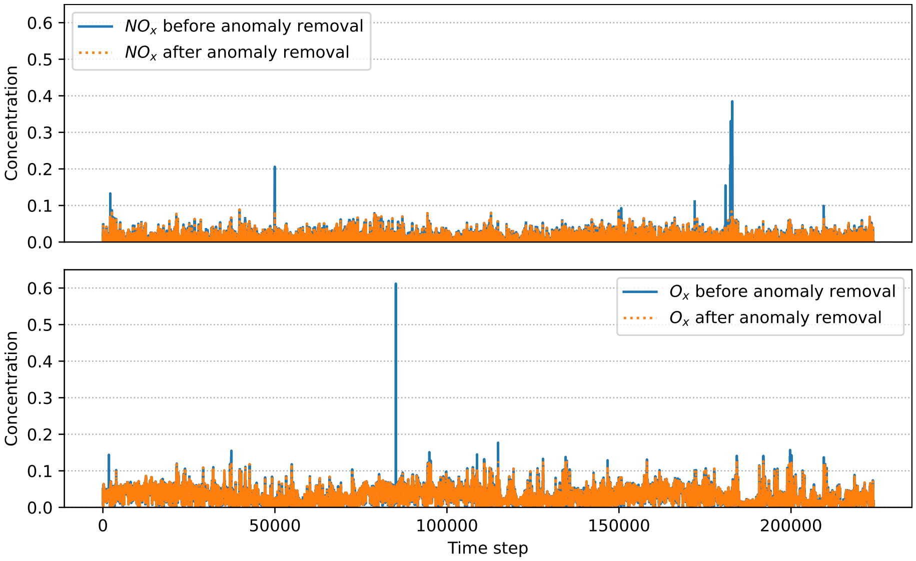

Fig. 6. Anomaly removal

A missing value filling technique was employed to address any gaps in the dataset and ensure a comprehensive and continuous time series. Specifically, missing values within the air pollution data were imputed by adopting an approach that involved substituting them with observations from the corresponding hour. This method aimed to preserve the temporal integrity of the dataset, acknowledging the intrinsic temporal patterns inherent in air quality data. By leveraging values from the same hour for imputation, the imputed data retained the temporal context, which is crucial for accurate and meaningful analyses in the subsequent stages of the air pollution forecasting model.

Min-max Scale: A Min-Max scaling step was applied to normalize the air pollution data, ensuring that all features were proportionally scaled between a predefined minimum and maximum range. This normalization enhances the convergence and stability of machine learning models by preventing one variable from dominating the others due to differences in scale.

4.2 Traffic Flow Dataset

The traffic dataset used in this study was released by R. Jiang et al. [17]. The dataset includes data on traffic speed recorded at 10-minute intervals for 1843 expressway road links in Tokyo, Japan, over a span of three months (from October 2021 to December 2021).

In Figure 7, the hourly average of vehicle speeds across all road links is depicted. A consistent pattern is evident across all three time periods (October, November, and December). During peak hours, particularly between 7 AM to

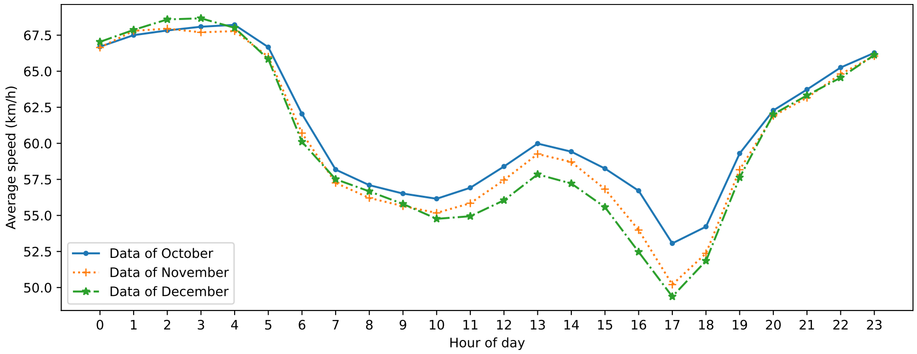

Fig. 7. Hourly average speed

9 AM and 5 PM to 6 PM, vehicle speeds tend to decrease. Conversely, during late-night hours, when traffic volumes are lower, speeds increase noticeably.

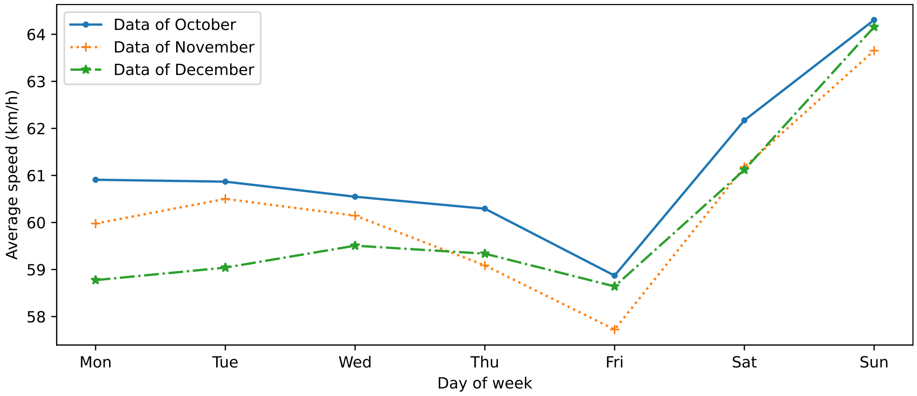

Fig. 8. Day of week average speed

In Figure 8, the average vehicle speed by day of the week across all road links is illustrated. A consistent pattern is evident across all three time periods (October, November, and December). On weekdays, when traffic volume is higher due to increased vehicular activity, the overall flow speed decreases. Conversely, during weekends, when there are fewer vehicles on the road, speeds tend to increase.

Figure 9 displays the actual speed for the first 1000 time steps. Upon comprehensive examination of the entire dataset, we confirmed the absence of any missing or abnormal values. Consequently, neither anomaly removal nor data imputation procedures were deemed necessary for this traffic dataset. Additionally, akin to the approach adopted for the air pollutant dataset, a Min-Max scaling

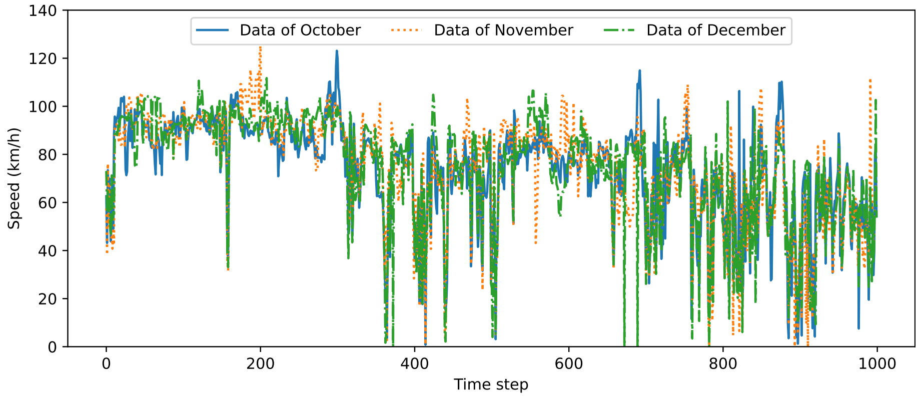

Fig. 9. Traffic speed anomaly removal

procedure was employed to normalize the speed data. This step ensures that all values are proportionally scaled within a predefined minimum and maximum range.

4.3 Training Deep Neural Networks

The task of forecasting air pollution is addressed as supervised machine learning problem, where the forecasting function $\mathbb{F}$ in Equation 1 is approximated by neural network models. In the comprehensive evaluation of forecasting models, a comparative study was conducted on four distinct DNN models: STGCN [6], STTN [11], TrafficSCINet [16], and the newly proposed STGTN.

For the air pollutant forecasting task, data from the years 2017 and 2018 were utilized to develop forecasting models, with the data from the year 2019 reserved exclusively for performance evaluation and comparison. Similarly, for the traffic flow forecasting task, data from the months of October and November were employed for model development, while the data from the month of December was held out for performance evaluation and comparison.

The assessment aimed to discern the forecasting performance of these models across varying forecasting horizons denoted as 1, 2, and 3 future steps (f value) using the historical data samples with a sequence length of 24 historical time steps. All models were developed in the same environment, described in Table 1. Furthermore, in the implementation of our proposed model STGTN, we utilized the Deep Graph Library (DGL), a high performance and scalable Python package for deep learning on graphs data [8].

5 Results and Discussion

The test results are derived from the evaluation of the forecasting models on the dedicated test datasets. Two widely used metrics, Root Mean Squared Error

Operating System	Ubuntu 20.04
Python version	3.8.16
Deep learning framework	Pytorch 1.13.1
Number of training epochs	200
Learning rate	0.001
Optimization algorithm	Adam [4]

Table 1. Setting of development environment

(RMSE) and Mean Absolute Error (MAE), were employed to quantitatively compare the accuracy of predictions. The detailed performance metrics provided in the comprehensive Table 2 and Figure 10.

Dataset		Air pollution				Traffic flow	
Prediction target		NO_x		O_x		$Speed$	
Model	Horizon	RMSE	MAE	RMSE	MAE	RMSE	MAE
STGCN	1	**0.0430**	0.0245	0.0470	0.0310	0.0546	0.0365
STTN	1	0.0479	0.0286	0.0596	0.0438	0.0532	0.0354
STGTN	1	0.0436	**0.0244**	0.0452	**0.0284**	**0.0529**	**0.0349**
TrafficSCINet	1	0.0435	0.0248	**0.0444**	0.0284	0.0548	0.0368
STGCN	2	0.0555	0.0339	0.0652	0.0474	0.0593	0.0392
STTN	2	0.0577	0.0344	0.0715	0.0535	0.0577	0.0379
STGTN	2	**0.0547**	**0.0316**	0.0599	**0.0414**	0.0577	0.0378
TrafficSCINet	2	0.0548	0.0317	**0.0597**	0.0414	**0.0568**	**0.0377**
STGCN	3	0.0620	0.0393	0.0775	0.0581	0.0622	0.0411
STTN	3	0.0640	0.0408	0.0831	0.0626	0.0608	0.0397
STGTN	3	**0.0606**	0.0366	**0.0703**	**0.0508**	0.0611	0.0399
TrafficSCINet	3	0.0607	**0.0363**	0.0716	0.0515	**0.0582**	**0.0385**

Table 2. Forecasting performance comparison

5.1 Air Pollution Forecasting

Across all forecasting horizons and pollutants, we observe variations in the RMSE and MAE values among different models. Notably, the STGTN model consistently demonstrates competitive performance, showcasing relatively lower RMSE and MAE values compared to other models in most cases. This suggests the effectiveness of the proposed STGTN with a dual attention mechanism in capturing the intricate spatial-temporal dynamics of the data. The performance of the TrafficSCINet model is slightly inferior to that of STGTN. Models STTN and STGCN show moderate performance across different scenarios, with varying levels of accuracy in capturing the underlying patterns in the data.

As the forecasting horizon increases from 1 to 3 hours, we observe a general trend of higher RMSE and MAE values across all models and pollutants.

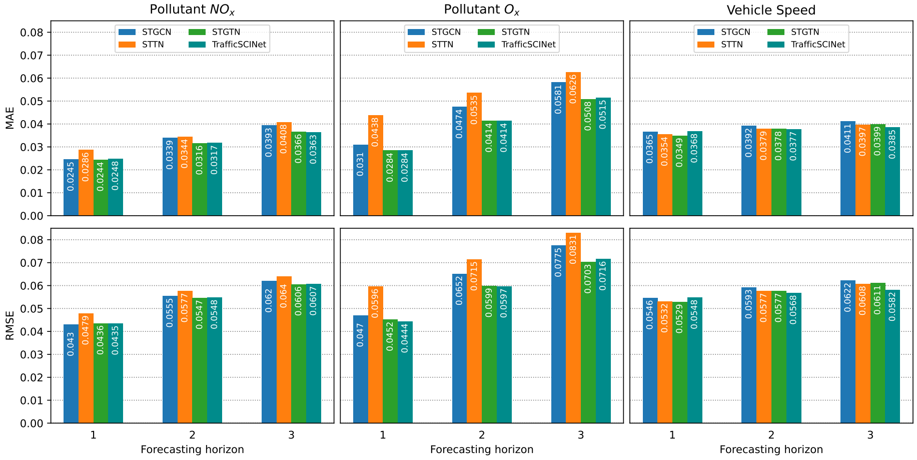

Fig. 10. Performance comparison

This indicates that predicting air pollutant concentrations becomes increasingly challenging as the forecasting time frame extends further into the future. The magnitude of increase in RMSE and MAE values varies across models.

Across all models and forecasting horizons, we observe that the concentrations of NO_x are generally higher compared to O_x. This discrepancy in magnitudes is evident in both RMSE and MAE values for the two pollutants. The performance of models vary between forecasting NO_x and O_x concentrations.

The relatively higher RMSE and MAE values for Ox compared to NO_x across models and forecasting horizons may indicate the inherent complexities in modeling O_x concentrations accurately.

5.2 Traffic Flow Forecasting

In the traffic flow forecasting task, both STGCN and STTN exhibit poor performance. TrafficSCINet, a model specifically tailored for this task, demonstrates the most favorable performance. Nevertheless, our proposed model also exhibits commendable performance, comparable to that of TrafficSCINet. Particularly noteworthy is STGTN's achievement of the highest performance in one-hour-ahead horizontal forecasting.

Similar to the air pollutant forecasting task, as the forecasting horizon extends from 1 to 3, we note a consistent trend of elevated RMSE and MAE values across all models. This trend suggests that forecasting traffic flow speed becomes progressively more challenging as the prediction window extends further into the future.

6 Conclusion

The STGTN model combining graph neural networks and transformer architectures effectively captures the complex spatial-temporal patterns present in the data. By integrating spatial dependencies and temporal dynamics in the dual attention mechanism, the model demonstrates its efficacy in addressing the challenges of spatial-temporal data analysis. Experimental study shows that, in the task of traffic flow forecasting, our model achieves comparable performance to the leading model tailored specifically for traffic flow forecasting, TrafficSCINet. Conversely, in air pollution forecasting, our proposed model demonstrates superior performance, underscoring its efficacy in addressing the complexities of this domain.

References

[1] Margaret A Oliver and Richard Webster. "Kriging: a method of interpolation for geographical information systems". In: *International Journal of Geographical Information System* 4.3 (1990), pp. 313–332.

[2] Richard A Bilonick. *An introduction to applied geostatistics.* 1991.

[3] Christopher Dienes and Alexander Aue. "On-line monitoring of pollution concentrations with autoregressive moving average time series". In: *Journal of Time Series Analysis* 35.3 (2014), pp. 239–261.

[4] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[5] Akhil Kadiyala and Ashok Kumar. "Vector time series-based radial basis function neural network modeling of air quality inside a public transportation bus using available software". In: *Environmental Progress & Sustainable Energy* 36.1 (2017), pp. 4–10.

[6] Bing Yu, Haoteng Yin, and Zhanxing Zhu. "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting". In: *arXiv preprint arXiv:1709.04875* (2017).

[7] Mohamad Sakizadeh, Mohamed MA Mohamed, and Harald Klammler. "Trend analysis and spatial prediction of groundwater levels using time series forecasting and a novel spatio-temporal method". In: *Water Resources Management* 33 (2019), pp. 1425–1437.

[8] Minjie Wang et al. "Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks". In: *arXiv preprint arXiv:1909.01315* (2019).

[9] Imad-Eddine Bouznad et al. "Trend analysis and spatiotemporal prediction of precipitation, temperature, and evapotranspiration values using the ARIMA models: case of the Algerian Highlands". In: *Arabian Journal of Geosciences* 13.24 (2020), p. 1281.

[10] Vijay Prakash Dwivedi and Xavier Bresson. "A generalization of transformer networks to graphs". In: *arXiv preprint arXiv:2012.09699* (2020).

[11] Mingxing Xu et al. "Spatial-temporal transformer networks for traffic flow forecasting". In: *arXiv preprint arXiv:2001.02908* (2020).

[12] Lei Xu et al. "Spatiotemporal forecasting in earth system science: Methods, uncertainties, predictability and future directions". In: *Earth-Science Reviews* 222 (2021), p. 103828.

[13] Minhao Liu et al. "Scinet: Time series modeling and forecasting with sample convolution and interaction". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 5816–5828.

[14] Bo Zhang et al. "Deep learning for air pollutant concentration prediction: A review". In: *Atmospheric Environment* 290 (2022), p. 119347. ISSN: 1352-2310. DOI: `https://doi.org/10.1016/j.atmosenv.2022.119347`. URL: `https://www.sciencedirect.com/science/article/pii/S1352231022004125`.

[15] Wenjie Du et al. "Deciphering urban traffic impacts on air quality by deep learning and emission inventory". In: *journal of environmental sciences* 124 (2023), pp. 745–757.

[16] Kai Gong et al. "TrafficSCINet: An Adaptive Spatial-Temporal Graph Convolutional Network for Traffic Flow Forecasting". In: *International Conference on Intelligent Computing*. Springer. 2023, pp. 628–639.

[17] Renhe Jiang et al. "Spatio-temporal meta-graph learning for traffic forecasting". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 7. 2023, pp. 8078–8086.

[18] Hexiang Liu et al. "Spatiotemporal Adaptive Attention Graph Convolution Network for city-level Air Quality Prediction". In: (2023).

[19] Manuel Méndez, Mercedes G Merayo, and Manuel Núñez. "Machine learning algorithms to forecast air quality: a survey". In: *Artificial Intelligence Review* (2023), pp. 1–36.

[20] Shuai Sun and Haiping Zhang. "Flow-data-based global spatial autocorrelation measurements for evaluating spatial interactions". In: *ISPRS International Journal of Geo-Information* 12.10 (2023), p. 396.